

Rapport final SAE 2.03

Fait par **Oussama Houaidj, Salah Eddine Houaidj, Isshak Abdelali**
Date: **23 mars 2025**

Sommaire

- Rapport final SAE 2.03
 - Date: **23 mars 2025**
- Sommaire
- Introduction
- Semaine 1
 - Installation de la VM
 - Questions ?
 - 1. Qu'est-ce qu'un fichier ISO bootable ?
 - 2. Qu'est-ce que MATE ? GNOME ?
 - 3. Qu'est-ce qu'un serveur web ?
 - 4. Qu'est-ce qu'un serveur SSH ?
 - 5. Qu'est-ce qu'un serveur mandataire ?
 - 6. Qu'est-ce que le Projet Debian ? D'où vient le nom Debian ?
 - 7. Les durées de prise en charge des versions Debian
 - 8. Durée de fourniture des mises à jour de sécurité
 - 9. Combien de versions sont maintenues par Debian ?
 - 10. D'où viennent les noms de code des distributions Debian ?
 - 11. Combien et quelles architectures sont prises en charge par Debian Bullseye ?
 - 12. Le premier et dernier nom de code utilisé par Debian
- Semaine 2
 - Apprentissage Markdown
 - À la semaine prochaine pour la semaine 3 !
- Semaine 3
 - Configuration globale de Git
 - Questions ?
 - 1.1. Qu'est-ce que le logiciel gitk ? Comment se lance-t-il ?
 - 1.2. Qu'est-ce que le logiciel git-gui ? Comment se lance-t-il ?
 - 2.1. Pourquoi avez-vous choisi ce logiciel ?
 - 2.2. Comment l'avez-vous installé ?
 - 2.3. Comparaison entre les différents outils
- Semaine 4
 - Introduction
- Installation de Gitea
 - Question ?
 - 1. Qu'est-ce que Gitea ?
 - 2. À quels logiciels bien connus dans ce domaine peut-on le comparer ?
 - 3. Qu'est-ce qu'un "fork" dans le domaine du développement logiciel ?

- 4. De quel logiciel Gitea est-il le fork ? Ce logiciel existe-t-il encore ?
 - Suite installation Gitea
- Fin

Introduction

Bonjour a vous et bienvenue dans notre rapport !

Nous sommes heureux de vous présenter le travail que nous avons réalisé dans le cadre de la ==SAE2.03==. Ce projet nous a permis de mettre en pratique nos compétences apprises tout au long de la SAE.

Nous espérons que vous apprécierez notre travail !

Semaine 1

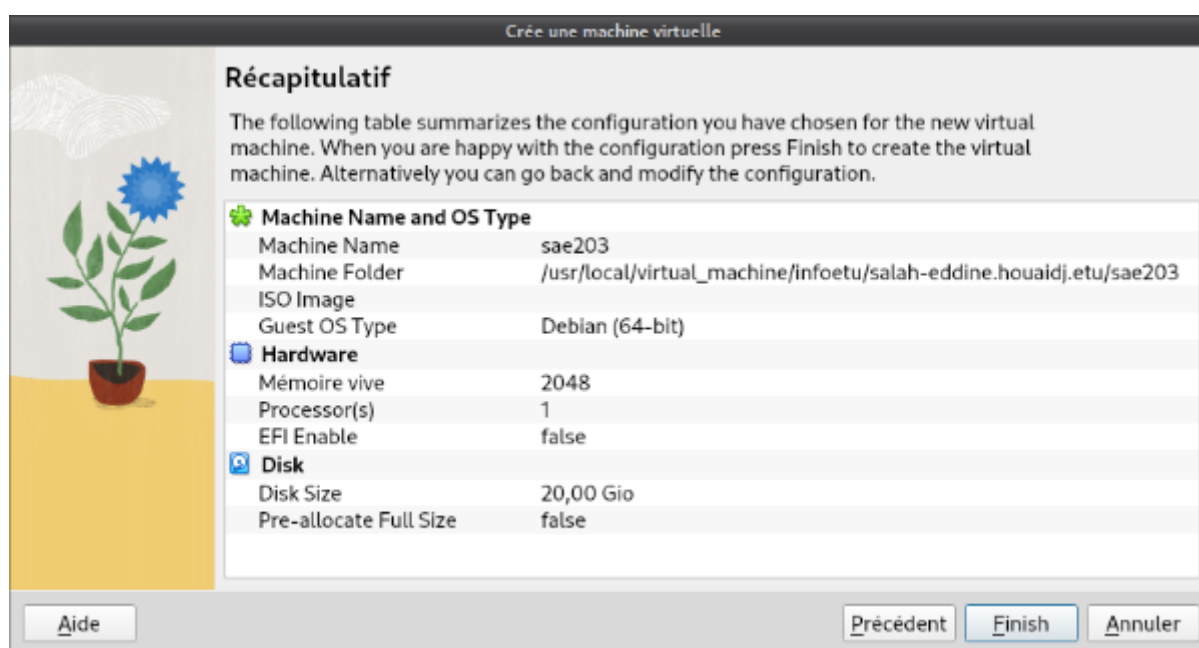
Installation de la VM

Pour l'installation de la VM, j'ai dans un premier temps conçu celle-ci en faisant bien attention aux paramètres demandés lors de la semaine 1, à savoir **2048 Mo de mémoire** et **20 Go de stockage**. Une fois la machine configurée comme l'énoncé le demandait, j'ai alors dû me procurer un ISO de la version 11 de Debian.

Alors plus à l'aise grâce à la SAE 1.03 effectuée il y a de cela plusieurs mois, j'ai alors décidé d'effectuer de nouveau une installation manuelle via une image ISO. J'ai alors sélectionné "Install" et j'ai commencé mon installation en faisant bien attention à remplir les mêmes caractéristiques que dans l'énoncé.

Il est important de rappeler que lorsqu'on choisit d'effectuer l'installation de cette manière, tout s'effectue via le clavier (espace pour sélectionner/désélectionner, etc.).

Ensuite, il suffit de suivre les instructions indiquées dans le document de la semaine 1 et nous devrions finir par obtenir une VM avec ces caractéristiques.



Étant donné que l'installation prend un temps considérable, j'ai donc commencé à répondre à quelques questions. Pour la plupart des réponses, j'ai pu les obtenir directement sur le site officiel de Debian.

Une fois l'installation terminée, je me suis connecté avec les identifiants que j'avais choisis. J'ai lancé un terminal avec cette commande :

```
CTRL+Alt+T
```

Ensuite j'ajoute mon User au groupe sudo pour avoir les **droits sudo** avec ces commandes :

```
su -(user)
usermod -aG sudo user
```

C'est tout ce qu'il fallait faire pour notre VM. Passons maintenant aux questions de la semaine 1.

Questions ?

1. Qu'est-ce qu'un fichier ISO bootable ?

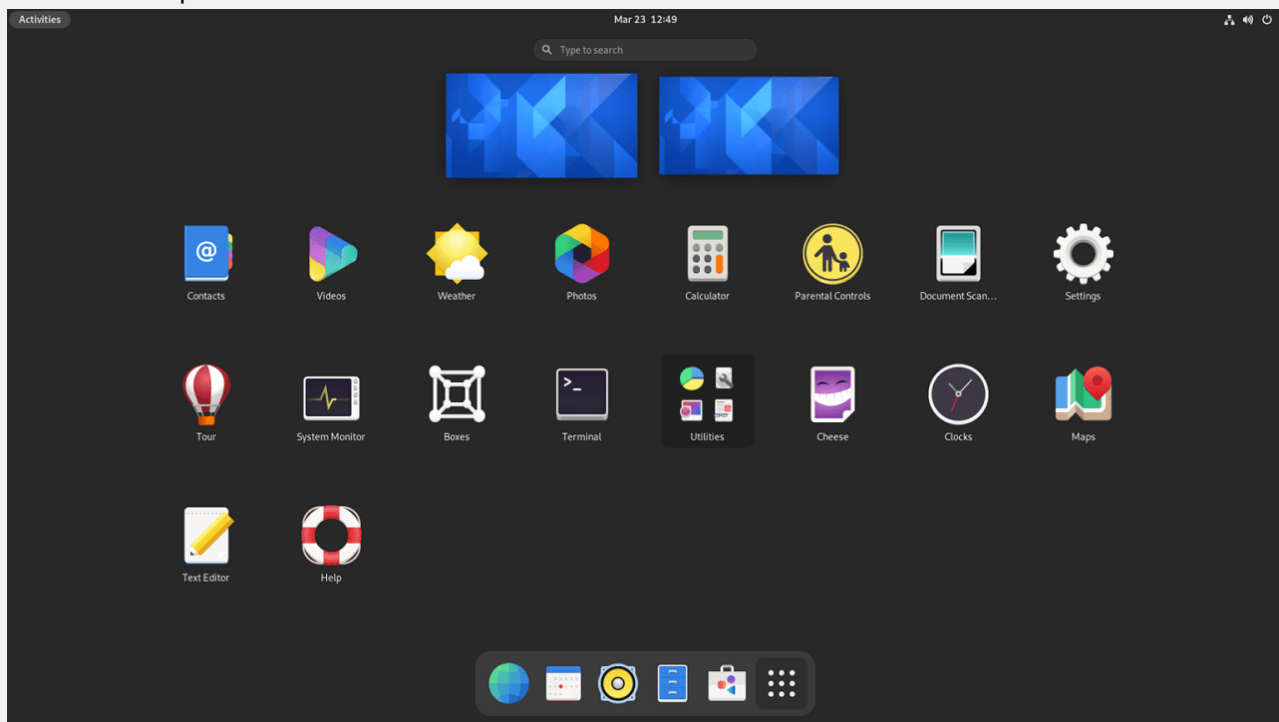
Un **fichier ISO** est une copie numérique exacte d'un disque, comme un CD, un DVD ou un Blu-ray. Lorsqu'il est "*bootable*", cela signifie qu'il peut être utilisé pour démarrer un ordinateur. Ces fichiers sont souvent utilisés pour installer des systèmes d'exploitation ou des logiciels. Pour les utiliser, on les copie généralement sur une clé USB, de manière à ce qu'on puisse démarrer l'ordinateur avec cette clé et installer ce qui est contenu dans l'ISO.

2. Qu'est-ce que MATE ? GNOME ?

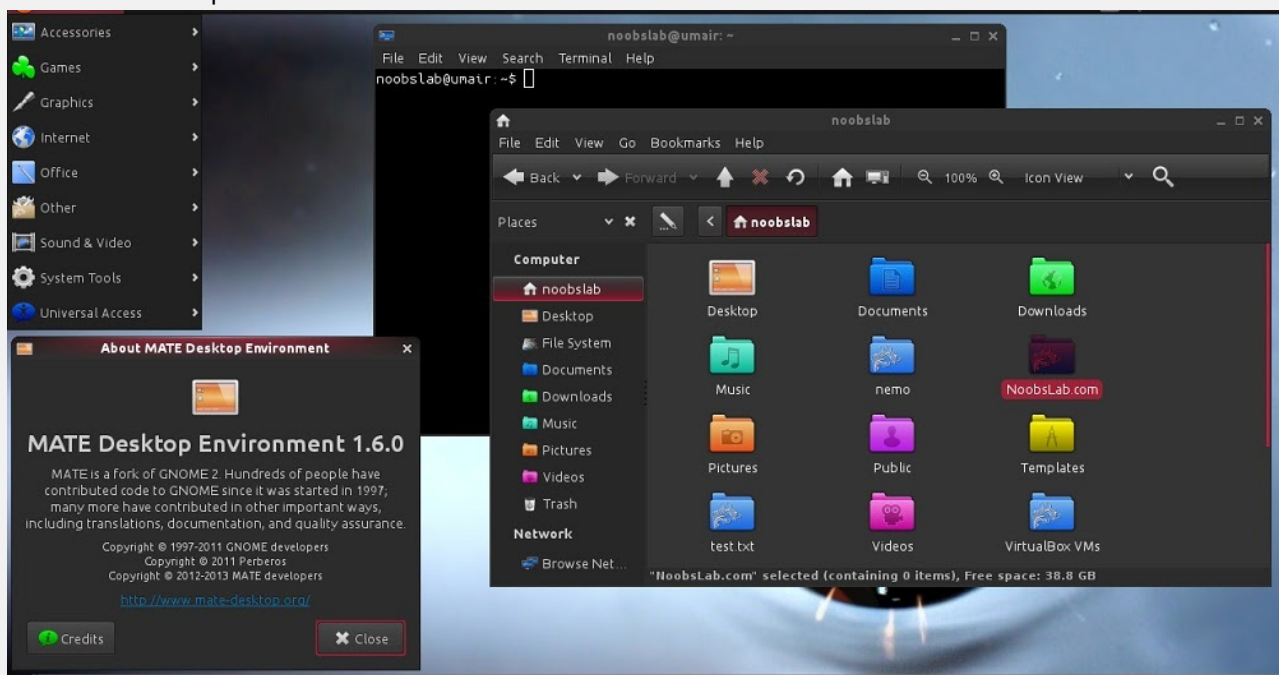
MATE et **GNOME** sont des types de bureaux utilisés sur des ordinateurs qui tournent avec des systèmes comme Linux.

- **GNOME** : C'est un bureau moderne et populaire, connu pour son interface simple et facile à utiliser. Il est souvent utilisé par des systèmes comme Ubuntu. Son but est de rendre l'usage de l'ordinateur plus rapide et agréable avec des outils pratiques et une interface claire.
- **MATE** : C'est un bureau qui ressemble à l'ancien GNOME avant qu'il ne change. Il est conçu pour ceux qui préfèrent une apparence plus traditionnelle, avec un menu et une interface familière. MATE est aussi moderne tout en gardant un style classique.

Voici un exemple d'environnement GNOME :



Et Voici un exemple d'environnement Mate :



3. Qu'est-ce qu'un serveur web ?

Un **serveur web** est un logiciel ou un matériel dédié qui sert du contenu web aux utilisateurs via le protocole HTTP ou HTTPS. Lorsqu'un utilisateur saisit une URL dans son navigateur, ce dernier envoie une requête au serveur web correspondant, qui répond en fournissant les ressources demandées, telles que des pages HTML, des images ou des fichiers. Les serveurs web les plus couramment utilisés incluent Apache, Nginx et Microsoft Internet Information Services (IIS).

4. Qu'est-ce qu'un serveur SSH ?

Les **serveurs SSH** (Secure Shell) sont des protocoles réseaux qui permettent d'établir des connexions sécurisées entre deux ordinateurs, généralement pour accéder à un système distant en ligne de commande.

Un serveur SSH est le logiciel qui implémente ce protocole sur la machine distante, permettant aux utilisateurs de se connecter de manière sécurisée pour exécuter des commandes, transférer des fichiers ou gérer le système. OpenSSH est l'une des implémentations les plus répandues de ce protocole.

5. Qu'est-ce qu'un serveur mandataire ?

Un **serveur mandataire** est un intermédiaire entre ton navigateur ou un client, et un site web. Quand tu veux accéder à une page, le proxy reçoit ta demande, la transmet au site web, puis te renvoie la réponse. Les proxys sont utilisés pour plusieurs raisons, comme améliorer la sécurité, contrôler l'accès à Internet, accélérer le chargement des pages en enregistrant des copies des contenus, ou cacher ton adresse IP réelle pour préserver ta confidentialité.

6. Qu'est-ce que le Projet Debian ? D'où vient le nom Debian ?

- Le **Projet Debian** est un groupe mondial de bénévoles qui travaillent ensemble pour créer un système d'exploitation libre, connu sous le nom de Debian GNU/Linux.
- Le nom Debian vient des prénoms de Debra, la compagne de Ian Murdock (le fondateur du projet), et de son propre prénom, formant ainsi "Debian".

7. Les durées de prise en charge des versions Debian

Il existe trois types de durées de prise en charge pour les versions de Debian :

- **Durée minimale** : Chaque version stable de Debian est supportée pendant 3 ans, avec des mises à jour de sécurité et des corrections de bugs majeurs.
- **Support Long Terme (LTS)** : Après les 3 ans de support initial, une version stable bénéficie de 2 années supplémentaires de support via le projet LTS, ce qui porte le total à 5 ans.
- **Support Long Terme Étendu (ELTS)** : Si une version nécessite encore du support après 5 ans, le projet ELTS propose jusqu'à 5 années supplémentaires, ce qui porte le total à 10 ans. Ce service est géré par Freexian et n'est pas un projet officiel de Debian.

8. Durée de fourniture des mises à jour de sécurité

Les **mises à jours de sécurtié** sont fournies pendant les ==3== premières années de support officiel. Ensuite, le projet LTS prend le relais pendant ==2== années supplémentaires, suivi, si nécessaire, par le projet ELTS qui offre jusqu'à ==5== années de support additionnel. Cela permet d'avoir jusqu'à ==10== ans de mises à jour de sécurité pour une version donnée.

9. Combien de versions sont maintenues par Debian ?

On en compte trois que ceux sont ci:

- **Stable** : Version actuelle recommandée pour la production.
- **Testing** : Future version stable en cours de test.
- **Unstable (Sid)** : Version en développement continu, également appelée "Sid".

10. D'où viennent les noms de code des distributions Debian ?

Les **noms de codes** des versions de Debian sont inspirés des personnages du film d'animation "Toy Story" de Pixar. Par exemple, "Buzz" pour Debian 1.1 et "Rex" pour Debian 1.2.

11. Combien et quelles architectures sont prises en charge par Debian Bullseye ?

Debian 11 **Bullseye** prend en charge plusieurs architectures, notamment :

- amd64 (PC 64 bits)
- arm64 (ARM 64 bits)
- armel (ARM EABI)
- armhf (ARM Hard Float)
- i386 (PC 32 bits)
- mips (MIPS (Big Endian))
- mips64el (MIPS 64 bits (Little Endian))
- mipsel (MIPS (Little Endian))
- ppc64el (PowerPC 64 bits Little Endian)
- s390x (IBM System z)

12. Le premier et dernier nom de code utilisé par Debian

Premier nom de code utilisé :

- **Buzz** : Le premier nom de code utilisé fut "Buzz" pour Debian 1.1.

Quand a-t-il été annoncé ?

- **Annonce** : Debian 1.1 "Buzz" a été annoncée le 17 juin 1996.

Quel était le numéro de version de cette distribution ?

- **Verson** : Le numéro de version était 1.1.

Un bon nombre de question auxquels on a finit par répondre.

Cette semaine 1 fut éprouvante.

Semaine 2

Apprentissage Markdown

Durant la semaine 2, aucun travail spécifique ne nous a été donné, mais une tâche importante nous a été confiée : choisir un langage de balisage léger et l'apprendre. Nous avons le choix entre **AsciiDoc** et **Markwon**. Après réflexion, notre groupe a opté pour **Markdown**.

Nous avons donc profité de cette période pour nous concentrer sur l'apprentissage des bases de **Markdown**. Nous avons exploré les principales fonctionnalités, telles que la création de titres, de listes, l'ajout de liens et l'intégration d'images dans un document. Ce langage étant simple et efficace, nous avons décidé de l'utiliser pour la rédaction de notre rapport.

En plus de cela, nous avons étudié l'histoire de **Markdown** ainsi que sa documentation officielle. Cela nous a permis de mieux comprendre son évolution et son rôle dans le monde de la rédaction technique.

Un autre point essentiel que nous avons abordé est l'exportation de documents **Markdown** en formats **PDF** et **HTML**, ce qui est crucial pour notre projet, car ces formats sont demandés pour la remise du rapport. Nous avons appris à utiliser des outils comme **Pandoc** pour effectuer ces conversions.

Ainsi, bien que cette semaine ait été principalement dédiée à l'apprentissage, elle a été essentielle pour établir les bases de notre travail futur. L'apprentissage a été long et rigoureux, mais il nous permettra de réaliser un rapport structuré et bien présenté à l'aide de **Markdown**.

Remarque : Cette phase d'apprentissage a été cruciale pour nous familiariser avec un outil indispensable pour notre projet, et nous sommes désormais prêts à appliquer nos nouvelles compétences dans la rédaction du rapport final.

À la semaine prochaine pour la semaine 3 !

Semaine 3

Durant cette semaine, nous avons suivi les étapes d'installation et de configuration comme indiqué dans le document de **Git**, un logiciel libre utilisé pour la gestion de versions de projets. Nous avons également installé les outils graphiques associés à Git, comme **gitk** et **git-gui**, et comparé ces outils avec une autre application graphique choisie parmi une liste.

Configuration globale de Git

Pour commencer, nous avons configuré Git sur notre machine virtuelle avec Debian en suivant les étapes indiqués dans la semaine 3. Nous avons utilisé les commandes suivantes pour paramétrer Git sur notre compte user en prenant comme exemple un membre de notre groupe :

```
git config --global user.name "Oussama Houaidj"
```

```
git config --global user.email "oussama.houaidj31@gmail.com"
```

```
git config --global init.defaultBranch "master"
```

Pour installer gitk et git-gui, nous avons simplement utilisé la commande suivante dans le terminal de ma machine :

```
sudo apt install git-gui  
sudo apt install gitk
```

Questions ?

1.1. Qu'est-ce que le logiciel gitk ? Comment se lance-t-il ?

gitk est un outil graphique qui permet de visualiser l'historique des ==commits== dans un dépôt Git. Il offre une interface conviviale pour parcourir les modifications et branches d'un projet.

Il se lance avec la commande suivante dans un terminal :

```
gitk
```

1.2. Qu'est-ce que le logiciel git-gui ? Comment se lance-t-il ?

Pour ce qui est de **git-gui** c'est une interface graphique qui permet de réaliser des opérations Git comme le ==commit==, le ==staging== et la gestion des branches sans utiliser la ligne de commande. Pour le lancer on utilise la commande suivante dans le terminal :

```
git gui
```

2.1. Pourquoi avez-vous choisi ce logiciel ?

Notre choix s'est porté sur Gitg et voici pourquoi:

- **Le prix :**
Nous avons choisi **gitg** car il est totalement gratuit, ce qui nous arrange bien.
- **La facilité d'installation :**
L'installation est simple et rapide, ce qui nous a permis de l'installer sans rencontrer de difficultés majeures.
- **La facilité d'utilisation :**
gitg offre une interface claire et intuitive qui nous permet de gérer facilement nos projets Git. Il nous propose aussi des fonctionnalités graphiques simples pour visualiser l'historique des commits, sans avoir besoin de maîtriser la ligne de commande.
- **Adapté à nos besoins :**
En plus de sa gratuité et de sa simplicité d'installation, **gitg** répond parfaitement à nos attentes pour un outil léger et facile à utiliser, tout en restant performant pour des projets personnels comme cet SAE .

2.2. Comment l'avez vous installé ?

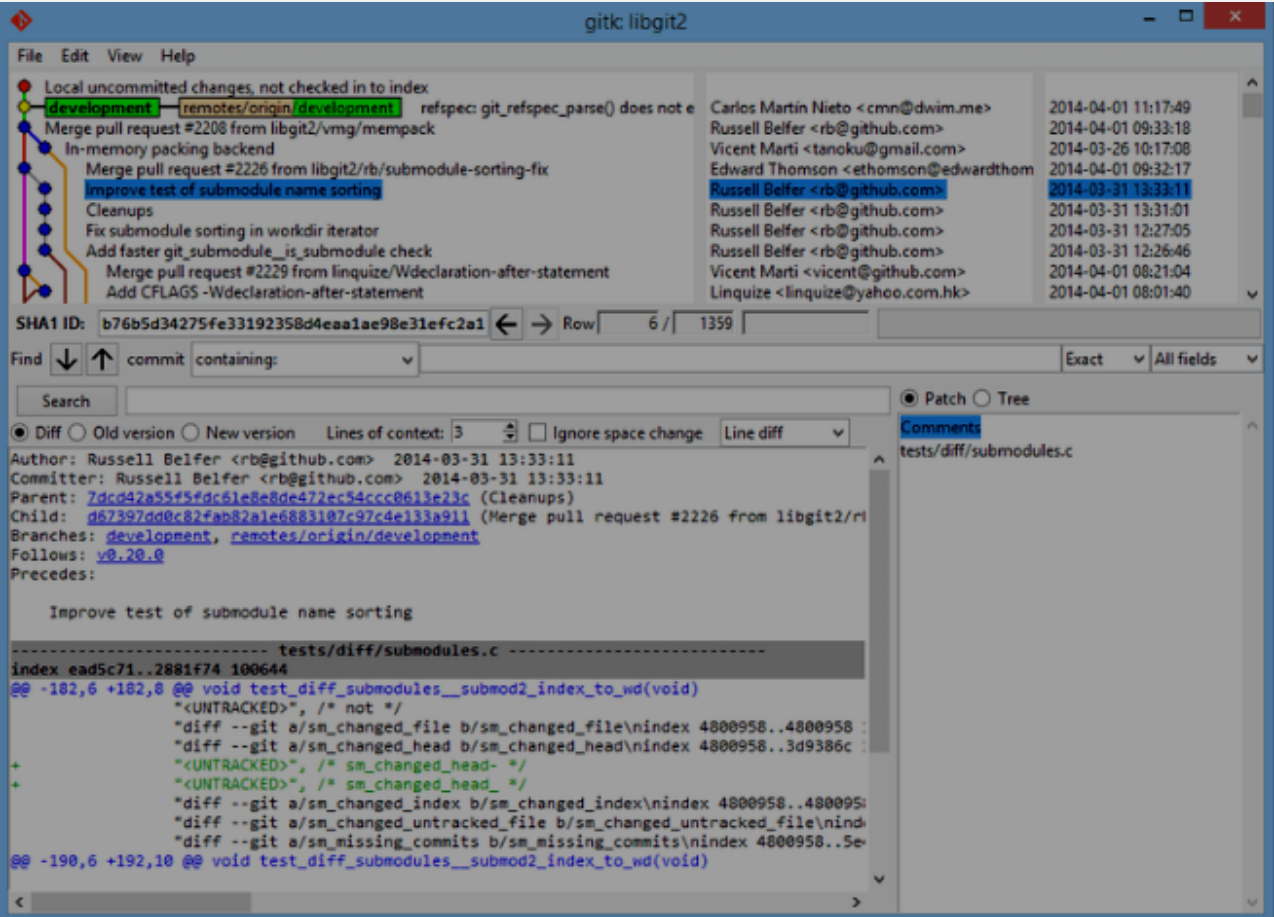
Comme dit précédemment, nous avons aussi choisi Gitg pour sa facilité d'installation. Voici la commande pour l'installer :

```
sudo apt install gitg
```


Et voici la commande pour le lancer:

gitg

Voici un exemple de l'interface de **gitg**



2.3. Comparaison entre les différents outils

Outil	Avantages	Inconvénients
gitk	Permet de visualiser graphiquement l'historique des commits	Interface datée et peu ergonomique
git-gui	Facile à utiliser pour faire des commits et gérer les fichiers et l'environnement de simulation	Moins complet pour la visualisation de l'historique
gitg	Interface simple et intuitive, bonne visualisation des commits, léger et rapide	Moins de fonctionnalités avancées, ne permet pas de gérer toutes les actions Git
Ligne de commande	Puissante et flexible, permet un contrôle total	Moins accessible aux débutants

Avec cette dernière question, notre travail pour cette semaine est donc terminé. Nous nous retrouverons la semaine prochaine pour la semaine 4, qui sera la dernière de notre rapport.

À la semaine prochaine !

Semaine 4

Introduction

On se retrouve pour la dernière semaine de notre **SAE 2.03**

Cette semaine nous avons procéder a l'installation et la configuration de ==**Gitea**= alors accrochez vous bien !

Installation de Gitea

Lors de l'installation de **Gitea** sur notre machine virtuelle, nous avons rapidement constaté un problème majeur : l'incapacité d'accéder au service Gitea depuis l'extérieur de la machine virtuelle. Cela était donc contraignant car ne nous permettait d'avoir accès que depuis un seul point. Le service, fonctionnant sur le port **3000** de la machine virtuelle, n'était pas accessible depuis la machine hôte. Cette situation était liée à la configuration du réseau en mode **NAT**, par défaut sur VirtualBox. Afin de résoudre ce problème, nous avons dû configurer une **redirection de port** pour permettre l'accès à Gitea depuis l'hôte.

```
https://www.virtualbox.org/manual/UserManual.html#networkingdetails
```

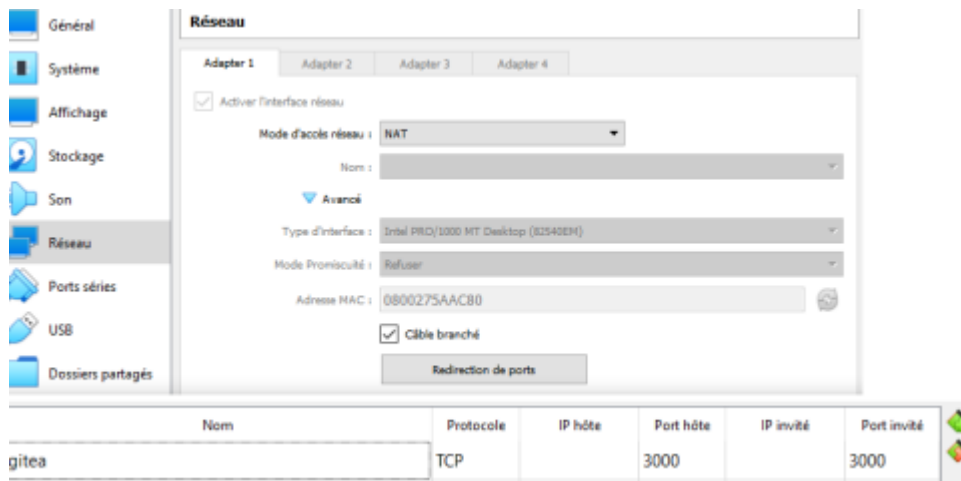
Ce lien présent dans le PDF de la semaine 4 et le tableau nous ont permis d'en apprendre plus et de comprendrem mieux ce qu'il fallait faire.

Nous avons donc configuré la **redirection de port** dans VirtualBox, ce qui a permis de rediriger les messages arrivant sur le port **3000** de la machine physique vers le port **3000** de la machine virtuelle. Grâce à cela, nous avons pu accéder à Gitea depuis notre navigateur sur l'hôte en utilisant l'URL :

```
http://localhost:3000
```



Ce lien ne fonctionnera que depuis le navigateur de l'hôte, comme indiqué.



Voici une image qui nous montre la redirection du port qu'il nous fallait reconfigurer.

Question ?

1. Qu'est-ce que Gitea ?

Gitea est une plateforme que l'on peut héberger soi-même pour aider au développement de logiciels. Elle offre plusieurs fonctionnalités pratiques, comme :

- **Hébergement Git** : pour gérer et stocker le code source d'un projet.
- **Revue de code** : pour permettre aux développeurs de vérifier et de discuter du code écrit par d'autres.
- **Collaboration en équipe** : pour travailler ensemble sur des projets logiciels.
- **Gestion des packages** : pour gérer les outils et bibliothèques utilisés dans un projet.

En somme **Gitea** facilite la vie.

2. À quels logiciels bien connus dans ce domaine peut-on le comparer ?

Les logiciels les plus connus et similaires à Gitea sont **GitHub** et **GitLab** que nous connaissons bien car beaucoup utilisent dans notre formation qu'est le BUT Info. Ces deux services permettent également de partager du code et de travailler en équipe sur des projets logiciels. Gitea offre les mêmes fonctionnalités, mais en étant auto-hébergé, ce qui signifie qu'on peut l'installer et le gérer soi-même sur ses propres serveurs.*

3. Qu'est-ce qu'un "fork" dans le domaine du développement logiciel ?

Un **fork** désigne la copie d'un projet déjà existant pour créer une nouvelle version différente. Cela permet à d'autres développeurs de travailler sur un projet sans affecter la version originale. Par exemple, on peut vouloir tester de nouvelles idées ou ajouter des fonctionnalités qui ne sont pas dans la version de base du projet.

4. De quel logiciel Gitea est-il le fork ? Ce logiciel existe-t-il encore ?

Gitea est un fork de **Gogs**, un autre service permettant de gérer des projets Git sur des serveurs personnels. **Gogs** est toujours disponible et continue d'être maintenu, donc les deux logiciels existent en parallèle, chacun avec ses propres spécificités.

Bien après avoir répondu aux questions, reprenons l'installation.

Suite installation Gitea

Pour la suite de l'installation, nous allons simplement suivre les 4 étapes de l'installation comme décrit dans le document.

Dans un premier temps, le fait que tout soit en anglais a compliqué la démarche. Lors d'une installation, il faut être précis dans les étapes et les commandes demandées, donc impossible de faire du "à peu près" en traduction.

Téléchargement de Gitea Je me suis alors rendu sur la page de téléchargement Gitea Download et nous avons choisi le fichier correspondant aux paramètres de ma machine, à savoir linux-amd64, comme recommandé par le service.

À noter qu'il est aussi possible de récupérer le fichier via le GitHub de Gitea (GitHub Releases). On y retrouve également toutes les corrections et mises à jour disponibles.

Ensuite, on installe Gitea avec la commande suivante :

```
wget -O gitea https://dl.gitea.io/gitea/1.23.5/gitea-1.23.5-linux-amd64
```

Puis, on déplace le fichier vers un emplacement plus approprié :

```
sudo mv gitea /usr/local/bin/
```

Mais pourquoi avons-nous choisi de déplacer le fichier dans **/usr/local/bin/** ? Pour trois raisons :

- **Pratique** : Une fois dedans, on peut taper gitea n'importe où dans le terminal, sans devoir naviguer dans les dossiers.
- **Propre** : /usr/local/bin/ est l'emplacement classique pour les logiciels installés manuellement, évitant ainsi d'avoir un fichier perdu quelque part.
- **Moins de galères**: Si on le laisse ailleurs, il faudra systématiquement faire ./gitea ou modifier le PATH pour l'exécuter globalement.

Verifions maintenant la version avec la commande :

```
gitea --version
```

```
user@serveur:~$ gitea -version
Gitea version 1.23.5 built with GNU Make 4.3, go1.23.7 : bindata, sqlite, sqlite_unlock_notify
```

Mais avant cela, il faut s'accorder les droits d'exécution avec une commande très simple:

```
sudo chmod u+x /usr/local/bin/gitea
```

À partir de là, tout est bon !

Passons maintenant à la création de notre utilisateur Git.

Pour ce faire, nous allons utiliser cette commande qui nous permettra de créer l'utilisateur:

```
adduser \  
  --system \  
  --shell /bin/bash \  
  --gecos 'Git Version Control' \  
  --group \  
  --disabled-password \  
  --home /home/git \  
git
```

```
user@serveur:~$ sudo adduser --system --shell /bin/bash --gecos 'Git Version Control' --group --disabled-password  
home /home/git git  
[sudo] Mot de passe de user :  
Ajout de l'utilisateur système « git » (UID 115) ...  
Ajout du nouveau groupe « git » (GID 122) ...  
Ajout du nouvel utilisateur « git » (UID 115) avec pour groupe d'appartenance « git » ...  
Création du répertoire personnel « /home/git »...
```

Voici une image de la commande.

Passons en mode root pour la suite :

```
su -
```

Puis il suffit de **copier-coller** les commandes demandées.

Pour **démarrer** Gitea, on utilise la commande :

```
gitea web
```

Si tout fonctionne correctement, Gitea se lance et affiche les logs dans le terminal. Ensuite, on accède à l'interface web. Avant cela, il faut récupérer l'adresse IP de notre VM avec la commande :

```
ip a
```

Cette IP nous permettra d'accéder à Gitea via un navigateur, dans notre cas, Firefox.

```
http://localhost:3000/
```

Nous permet d'accéder a Gitea Web

Une fois sur l'interface, il faut suivre scrupuleusement les indications données dans les consignes.

127.0.0.1:3000

Paramètres de Messagerie

Paramètres Serveur et Tierce Parties

▼ Paramètres de compte administrateur

La création d'un compte administrateur est facultative. Le premier utilisateur enregistré deviendra automatiquement un administrateur le cas échéant.

Nom d'utilisateur administrateur

Courriel

Mot de passe

Confirmez le mot de passe

Ces options de configuration seront écrites dans : `/usr/local/bin/custom/conf/app.ini`

Nous avons rencontré de nombreuses difficultés avec cette étape, notamment à cause d'un problème récurrent lié aux droits. Nous avons eu à plusieurs reprises des **"permission denied"**, ce qui a été éprouvant, mais j'ai finalement réussi grâce à la commande **chmod**, que j'ai adaptée à mes besoins.

Revérifions notre version de Gitea :

```
gitea --version
```


```
user@serveur:~$ gitea -version
Gitea version 1.23.5 built with GNU Make 4.3, go1.23.7 : bindata, sqlite, sqlite_unlock_notify
```

Pour ce qui est de **mettre à jour** sans avoir à tout reconfigurer, cette commande est capable de répondre à nos besoins :

```
sudo wget -O /usr/local/bin/gitea
```

Un nouveau petit coup de **gitea -version** pour vérifier tout ça et c'est nickel !




Tout est bon, Gitea est installé et prêt à être configuré !

 Tickets

Demandes d'ajout

Jalons

Explorateur

   admin

 gitea ▾



Aucune activité

Vous n'êtes actuellement abonné à aucun dépôt ou utilisateur et il n'y a donc aucun contenu à afficher. Les liens ci-dessous vous permettront d'explorer des dépôts ou des utilisateurs susceptibles de vous intéresser.

[Explorer des dépôts](#) · [Explorer des utilisateurs](#)

Dépôt

Organisation

Dépôts 0

+




Il n'y a pas encore de dépôts.

Nous avons réalisé plusieurs tests pour vérifier si tout fonctionnait bien.

Nouveau dépôt

Un dépôt contient tous les fichiers d'un projet, ainsi que l'historique de leurs modifications. Vous avez déjà ça ailleurs ? [Migrez-le ici.](#)

Propriétaire *  gitea ▼
Certaines organisations peuvent ne pas apparaître dans la liste déroulante en raison d'une limite maximale du nombre de dépôts.

Nom du dépôt * test
Idéalement, le nom d'un dépôt devrait être court, mémorisable et unique. Vous pouvez personnaliser votre profil ou celui de votre organisation en créant un dépôt nommé « .profile » ou « .profile-private » et contenant un README.md.

Visibilité ☐ Rendre le dépôt privé
Seuls le propriétaire ou les membres de l'organisation, s'ils ont des droits, seront en mesure de le voir.

Description

Modèle

Jeu de labels pour les tickets

le test fonctionne et prouve alors que l'environnement est fonctionnel

Fin

Voici donc ce qui conclut notre travail pour cette dernière semaine et pour l'ensemble de cette SAE. Nous avons été très heureux de pouvoir en apprendre davantage sur les réseaux, les différents outils Git, mais aussi d'apprendre un nouveau langage, le Markdown.

Merci d'avoir suivi notre rapport. Nous espérons que vous l'avez apprécié et que vous en garderez un bon souvenir. Ce projet touche désormais à sa fin, et nous sommes fiers d'avoir pu réaliser un rapport que nous estimons de qualité.

Au revoir !

